A genetic algorithm for slope stability analyses with concave slip surfaces using custom operators

Rafael Jurado-Piña and Rafael Jimenez

Heuristic methods are popular tools to find critical slip surfaces in slope stability analyses. A new genetic algorithm (GA) is proposed in this work that has a standard structure but a novel encoding and generation of individuals with custom-designed operators for mutation and crossover that produce kinematically feasible slip surfaces with a high probability. In addition, new indices to assess the efficiency of operators in their search for the minimum factor of safety (FS) are proposed. The proposed GA is applied to traditional benchmark examples from the literature, as well as to a new practical example. Results show that the proposed GA is reliable, flexible and robust: it provides good minimum FS estimates that are not very sensitive to the number of nodes and that are very similar for different replications.

Keywords: slope stability; evolutionary algorithms; heuristic methods; limit equilibrium; Spencer method

1. Introduction

Slope stability is a common geotechnical problem with significant practical and economic importance (Schuster 1996). Of the currently available methods for slope stability analyses, limit equilibrium is still, and will probably continue to be, the most commonly used by designers (Sarma and Tan 2006). In practice, engineers need to find the critical slip surface with the minimum factor of safety (FS). Therefore, it is natural to employ optimization methods. Initial efforts employed deterministic optimization—see Sarma and Tan (2006) for a review and references. However, even for relatively simple problems, the solution space is non-smooth and complex, with discontinuous functions and multiple minima; this makes such deterministic methods 'much more likely to find a false minimum than the true minimum' (McCombie and Wilkinson 2002).

Heuristic (or random search) methods—see for example Chen (1992), Greco (1996) and Bolton, Heymann, and Groenwold (2003)—have become a popular alternative (Taha, Khajehzadeh, and El-Shafie 2010). Genetic Algorithms (GAs) are heuristic algorithms that build on evolutionary biology and natural selection (Mitchell 1996; Goldberg 1989) to solve optimization problems.

For slope stability, (binary) GAs were initially employed for circular (McCombie and Wilkinson 2002) and non-circular (Zolfaghari, Heath, and McCombie 2005) slip surfaces, although, given their non-optimum results, more recent efforts employed alternative approaches—see for example

Cheng (2003), Cheng *et al.* (2007), Cheng, Li, and Chi (2007), Cheng (2007), Cheng *et al.* (2008a, 2008b), Kahatadeniya, Nanakorn, and Neaupane (2009) and Kang, Li, and Ma (2013). However, recent research suggests that, with adequate encodings and operators, GAs and evolutionary algorithms can become a reliable and efficient tool for slope stability analyses (Sun, Li, and Liu 2008; Li *et al.* 2010; Rickard and Sitar 2012). A new GA, with custom encoding of slip surfaces and GA operators, is proposed in this work to solve slope stability problems. Several examples are employed as benchmark cases, and new indices are proposed for assessing the performance of the GA.

2. Description of the genetic algorithm

2.1. *Structure of the algorithm*

An initial population is generated first and the fitness of its individuals is computed; then, building on modifications to Michalewicz (1996) proposed by Jong (1998), the next generations is obtained as follows:

(i) several parents are selected (with replacement) using selection probabilities based on their rankings (see Equation 1 below);
(ii) such selected parents are crossed or mutated to generate N_{off} offspring, which are inserted into the next generation;
(iii) using elitism (*i.e.* keeping the best N_{elit} solutions), N_{off} individuals are selected to be 'killed' from the parent population using 'inverse' ranking;
(iv) the surviving parents are transferred into the next generation.

This procedure is repeated until convergence, or until a given number of generations, N_{gen}, has occurred. The details of the proposed GA, which are specific to the slope stability problem analysed herein, are explained below.

2.2. *'Encoding' of geometrically feasible individuals*

Trial slip surfaces are open two-dimensional polygons with N vertices (or 'nodes'). Nodes are denoted as V_i, where $i = 1, \ldots, N$ indicates order within the slip surface, and nodes are numbered (from 1 to N) from 'top to bottom'. A trial slip surface ('individual') is completely defined with the coordinates of its nodes within an $\{x, y\}$ orthogonal reference system. The trial slip surface, S, can then be expressed using a matrix that includes, as columns, the vectors of coordinates of its vertices, as $S = [V_1, \ldots, V_N]$, where $V_i \equiv (x_i, y_i)^{\mathrm{T}}$, with $i = 1, \ldots, N$.

The location of nodes needs to fulfil some restrictions. For instance, the 'extreme' nodes (*i.e.* with $i = 1$ or $i = N$) must lie, within specific intervals, on the slope boundary. Such 'allowable extreme intervals', which are defined by the analyst, are denoted as $[P_{L1}, P_{U1}]$ and $[P_{LN}, P_{UN}]$ (see Figure 1).

Similarly, the 'interior' nodes (*i.e.* for $i = 2, \ldots, N-1$) must lie within an 'allowable search domain' defined by the designer. And, following a common assumption—see for example Malkawi, Hassan, and Sarma (2001), Sarma and Tan (2006), Cheng *et al.* (2007), Sun, Li, and Liu (2008), Kahatadeniya, Nanakorn, and Neaupane (2009) and Li *et al.* (2010)—only kinematically feasible, upwards-concave, trial slip surfaces are considered.

After locating V_1 and V_N, the V_1V_N segment is divided into $N-1$ equally-sized intervals. The 'interior' nodes lie on auxiliary lines, perpendicular to V_1V_N, that pass through the $N-2$ interval extremes, following a procedure that assures kinematic admissibility (see Section 2.3).

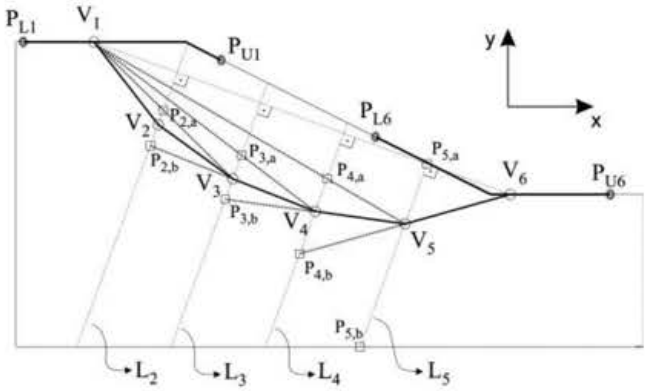


Figure 1. Encoding of individuals, and generation of slip surfaces of the initial population.

Other researchers have used similar encodings using vertical auxiliary lines (with constant or non-constant width)—see for example Bolton, Heymann, and Groenwold (2003), Cheng *et al.* (2008a) and Kang, Li, and Ma (2013). However, the approach proposed herein is preferred, since the best slope stability results 'are obtained by subdividing the soil mass such that the lengths of the base of slices, [...], are approximately equal, rather than using slices with equal widths' (Duncan and Wright 2005).

2.3. *Generation of the initial population*

A dynamic procedure that assures kinematic admissibility of generated slip surfaces is possible—see for example Cheng *et al.* (2008a). Figure 1 shows an example with $N = 6$ nodes. To agree with it, the discussion below mainly considers slip surfaces generated from 'bottom to top'; 'top to bottom' generation is formally equivalent, with the obvious changes of indices, and is not presented in detail.

The steps to produce slip surfaces for the initial population are as follows.

(1) The extreme vertices V_1 and V_N are randomly located within their 'allowable extreme intervals', defined by their corresponding lower and upper extremes, P_L and P_U.
(2) The V_1V_N segment is then used to produce $N-2$ parallel and equidistant auxiliary lines perpendicular to V_1V_N. They are referred to as L_i in Figure 1, with $i = 2, \ldots, N-1$.
(3) A 'flag' (either 0 or 1) is randomly generated to decide whether the slip surface will be generated from 'top to bottom' (*i.e.* for increasing i's) or vice versa. Next, for each L_i, a (dynamic) feasible interval $[P_{i,a}, P_{i,b}]$ is defined. The definition of feasible intervals is described below.
(4) To define the upper extreme of the feasible interval for the interior node generated first (*i.e.* in this example, $P_{N-1,a}$), either the intersection of its auxiliary perpendicular line, L_{N-1}, with the boundaries of the search domain or with the V_1V_N segment (whichever is lower), is used. Soil mechanics can be employed to estimate ranges of toe angles ($P_{N-1,b}$) and slope top angles. Once the feasible interval for V_{N-1} along L_{N-1} is defined, the V_{N-1} node is randomly located within such interval using a uniform probability distribution.
(5) The remaining nodes are generated iteratively with a procedure that assures kinematic admissibility of the resulting slip surface. Once a node (let's call it V_i) is located, the feasible interval for the next node (V_{i-1} for the 'bottom to top' example), within its auxiliary orthogonal line (L_{i-1}), results at its intersection with the following two lines (see Figure 1).

(a) The line segment that joins V_i with the 'final' node of the slip surface (V_1 for this example). The intersection of such a segment with the auxiliary perpendicular line (L_{i-1}) produces one extreme of the feasible region ($P_{i-1,a}$).
(b) The straight line that passes through node V_i and also through the previously generated node (V_{i+1} for 'bottom to top' generation). The intersection of such a line with the auxiliary perpendicular line (L_{i-1}) produces the other extreme of the feasible region ($P_{i-1,b}$).

If any (or both) such extreme intervals ($P_{i-1,a}$ or $P_{i-1,b}$) fall outside the 'allowable search domain', they are substituted by the intersection of L_{i-1} with the boundary of the 'allowable search domain'. Once the feasible interval for a node is defined, the node is randomly placed within the interval using a uniform probability distribution.

This procedure is repeated to generate all remaining interior nodes. Once the last internal node (V_2 in this case) has been generated, the last segment of the slip surface is obtained by simply joining it to the corresponding extreme node (V_1).

2.4. *Fitness computation and selection schemes*

Next, the 'goodness' (or 'fitness') of an individual, as well as how to select individuals for reproduction or mutation, are defined. The factor of safety (FS) arises as a natural measure of fitness: the lower the FS of a slip surface, the better it is as a solution. Factors of safety are computed using Spencer's method (Spencer 1967) as implemented in SLOPE8R (Duncan and Song 1984), with minor changes in the source code to improve convergence.

The performance of GAs, in terms of exploration and/or exploitation, can be controlled if the selection of individuals is based on their 'fitness ranking' (*i.e.* the individual with lowest FS is ranked 1, the second best FS is ranked 2, *etc.*). In rank-based selection, the probability of selecting an individual is (Michalewicz 1996):

$$p_k = c\, q\, (1-q)^{k-1}, \tag{1}$$

where p_k is the probability of selecting the individual with the kth fitness ranking, $q \in [0, 1]$ is a user-defined parameter, and c is obtained from the condition that all p_k probabilities must add up to one.

2.5. *'Recoding' of individuals*

Slip surfaces need to be 'recoded' in two situations:

(i) when the position(s) of one (or both) of the extreme nodes is (are) modified; and
(ii) when the number of nodes changes.

Figure 2 illustrates the first case. For a slip surface with N nodes, the 'recoding' operator produces equally-spaced auxiliary lines, L'_i ($i = 2, \ldots, N-1$), perpendicular to the segment joining the (possibly displaced) extreme nodes V'_1 and V'_N. The recoded interior nodes, V'_i, result at the intersection of L'_i with the original slip surface. The second type of recoding, which will be employed with the 'lumped' mutation operator discussed below, is conceptually equivalent, and will not be explained in detail.

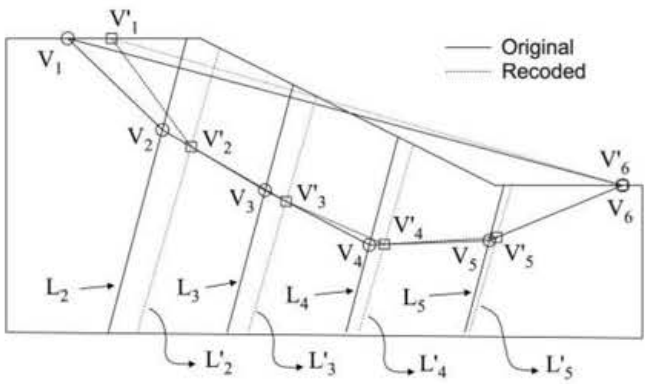


Figure 2. Illustration of the 'recoding' of a slip surface with displaced extreme points.

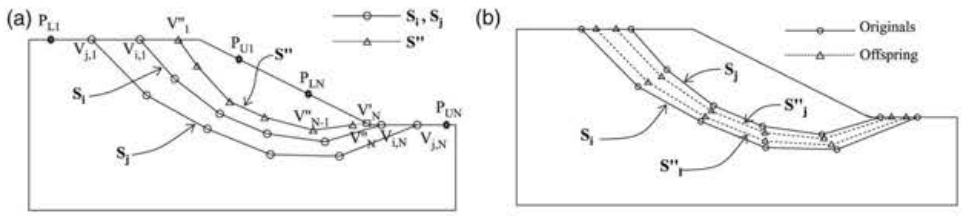


Figure 3. Description of (a) the heuristic crossover operator (not showing the final recoding operation) and (b) the arithmetic crossover operator.

3. Custom operators for slope stability problems

New and custom GA operators to work with slip surfaces encoded as indicated in Section 2.2 are presented next. When possible, operators are designed to ensure that they only generate kinematically feasible solutions.

3.1. *Crossover operators*

3.1.1. *Heuristic crossover*

Given two polygonal slip surfaces, S_i and S_j, with factors of safety $FS(S_i) \leq FS(S_j)$, the heuristic crossover operator produces one new polygonal as offspring. To that end, an auxiliary polygonal is initially computed, S'', as (see Figure 3(a)):

$$S'' = S_i + \xi(S_i - S_j), \tag{2}$$

where, here and in the rest of the discussion below, ξ is a random number from a $[0, 1]$ uniform distribution. In some cases, S'' will not fulfil the geometrical constraints; therefore, additional steps are required to generate a valid offspring, S', as follows (see Figure 3(a)).

(1) Once S'' is computed, it is checked whether the extreme nodes (V_1'' and V_N'') are within the corresponding 'allowable extreme intervals'. For $i = 1$ or $i = N$, several situations can occur:
   (a) if V_i'' is within the 'allowable extreme interval', it will be taken as the extreme of the offspring's polygonal S', *i.e.* $V_i' = V_i''$;
   (b) if V_i'' lies outside the 'allowable extreme interval', the S'' polygonal is 'intersected' by the boundary of the search domain to produce a new extreme node, V_i', as follows:

(i) if V_i'' lies outside the allowable search domain, V_i' is the intersection between S'' and the search domain boundary;

(ii) if V_i'' lies inside the allowable search domain, V_i' results from extending the segment of S'' adjacent to V_i'' until it intersects with the boundary of the search domain (see V_N' in Figure 3(a)).

If V_i' lies within the 'allowable extreme interval', then it is used as the extreme point of the offspring polygonal. Otherwise, a new S'' auxiliary polygonal is generated using Equation (2), until both extreme points are valid, producing nodes V_1' and V_N'.

(2) Using the new extreme vertices, V_1' and V_N', the auxiliary polygonal S'' is 'recoded' (Section 2.5). If the resulting polygonal S' is admissible—*i.e.* upwards-concave, with adequate extreme angles, *etc.*—it is taken as the offspring; otherwise, it is rejected and a new trial offspring is generated. If no valid offspring is obtained after a user-defined number of trials, the operator gives up and returns S_i as the offspring.

3.1.2. *Arithmetic crossover*

Given two parents, S_i and S_j, the arithmetic crossover operator computes two auxiliary polygonals, S_i'' and S_j'', as their linear combination:

$$\begin{aligned} S_i'' &= \xi S_i + (1-\xi) S_j \\ S_j'' &= \xi S_j + (1-\xi) S_i. \end{aligned} \tag{3}$$

Given S_i'' and S_j'' (see Figure 3(b)), their extreme nodes may lie outside the corresponding 'allowable extreme intervals'; in that case, they need to be 'corrected' using the heuristic crossover procedures. Then offspring resulting from this operator, S_i' and S_j', are obtained after recoding S_i'' and S_j'' (Section 2.5).

3.2. *Mutation operators*

3.2.1. *Uniform and non-uniform mutation of single nodes*

The uniform and non-uniform mutation of single nodes operate on non-aligned nodes (see Section 3.2.5 for a definition of 'aligned nodes'). The procedure is slightly different when the node is 'interior' (*i.e.* for $i = 2, \ldots, N-1$) from when it is 'extreme' (*i.e.* for $i = 1$ or $i = N$). Figure 4 illustrates an example, and the operator is further described below.

- *For 'interior' nodes,* the operator acts on a node, V_i, randomly selected from among the non-aligned nodes. Using V_i and geometrical considerations (Section 2.3), a feasible interval $[P_0, P_1]$

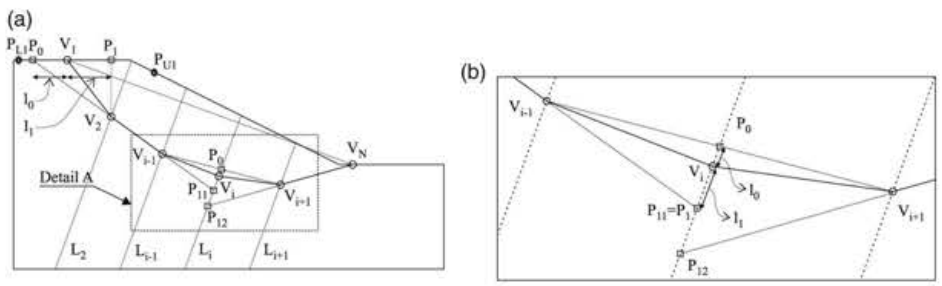


Figure 4. Description of the uniform and non-uniform mutation of single nodes. (a) Overall description and (b) Detail A.

on L_i is computed. P_0 is obtained as the intersection between L_i and the segment that joins the preceding and subsequent nodes, V_{i-1} and V_{i+1}. P_1 is defined as the point closer to V_i among two options: P_{11}, given by the intersection between L_i and the extension of the $V_{i-2}V_{i-1}$ segment; and P_{12}, given by the intersection between L_i and the extension of the $V_{i+1}V_{i+2}$ segment. (If $i = 2$, only P_{12} exists, and $P_1 = P_{12}$; similarly, if $i = N - 1$, only P_{11} exists and $P_1 = P_{11}$. See Figure 4.) If P_0 or P_1 are outside the 'allowable search domain', they are substituted by the intersection of L_i with its boundary. Using P_0 and P_1, two auxiliary segments, l_0 and l_1, are defined: l_0 joins V_i with P_0, whereas l_1 joins V_i with P_1 (see Figure 4(b)). The mutation operator displaces V_i along l_0 or l_1; with probability of selection proportional to their lengths. Using j to indicate the direction of mutation (*i.e.* $j = 0$ or $j = 1$), the mutated node, V_i', is computed as $V_i' = V_i + (P_j - V_i)\,\xi$ for the *uniform mutation operator*, and as $V_i' = V_i + (P_j - V_i) f(g)$ for the *non-uniform mutation operator*, where $f(g)$ is a 'damping function' given by $f(g) = \xi\,(1 - g/N_{\text{gen}})^{\epsilon}$, with $1 \leq g \leq N_{\text{gen}}$ being the generation number, and with ϵ being a predefined parameter that determines the degree of non-uniformity of the mutation. ($\epsilon = 6$ is used in this work.)

- *For 'extreme' nodes,* the procedure is similar, but with slight differences for extreme nodes. The mutation of V_1 is explained here; for V_N it is analogous and will not be discussed. The feasible interval for the extreme node $[P_0, P_1]$ is defined as follows (see Figure 4(a)): the 'exterior' interval limit (*e.g.* P_0 for V_1) results at the intersection of the adjacent segment's extension (*i.e.* V_2V_3) with the slope boundary; whereas the 'interior' limit (P_1 for V_1) results at the intersection with the slope boundary of a line traced from V_2 with a user-defined angle. (The search space can be reduced by considering Rankine's limit states, or any other physical constraints.) If P_0 or P_1 (or both) lie outside their 'allowable extreme interval', then it (they) is (are) moved to the nearest interval extreme. Using such a $[P_0, P_1]$ interval, the extreme node is mutated as before. In this case, however, since one of the extreme nodes has changed, the polygonal is 'recoded' (Section 2.5) to obtain an admissible offspring.

3.2.2. *Total non-uniform mutation*

This operator applies the non-uniform mutation operator (Section 3.2.1) to *all nodes* in the slip surface. Nodes are selected randomly (without replacement) and the surface is 'recoded' when one of the extreme nodes is mutated.

3.2.3. *Straight mutation*

Figure 5(a) illustrates the straight mutation operator. Its objective is to convert several (probably non-aligned) segments of the slip surface into one straight line. The ordinals, i and j (it is assumed that $i < j - 1$), of the two extreme nodes over which the straight mutation acts, are randomly chosen from the discrete set $\{1, \ldots, N\}$; then, the V_k nodes (with $k = i + 1, \ldots, j - 1$) that are located between V_i and V_j are transformed using $V_k' = V_i + (V_j - V_i)\,(k - i)/(j - i)$.

3.2.4. *'Extreme vertices' mutation*

The 'extreme vertices' mutation operator acts on both extreme nodes of the polygonal, V_i (with $i = 1$ and $i = N$), as follows (see Figure 5(b)):

(1) first, the 'allowable extreme interval', $[P_{Li}, P_{Ui}]$, corresponding to V_i is divided into two sub-intervals, $l_{i,1}$ and $l_{i,2}$;

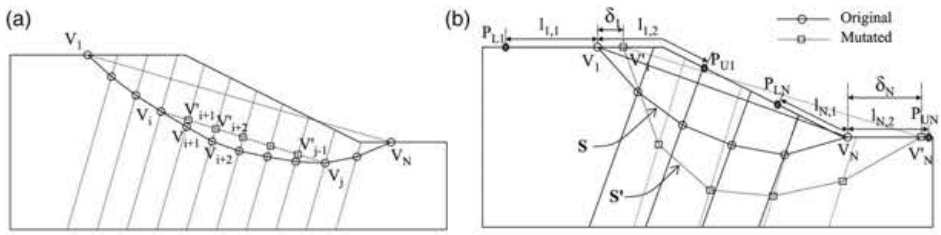


Figure 5. Description of (a) the straight mutation operator and (b) the 'extreme vertices' mutation operator.

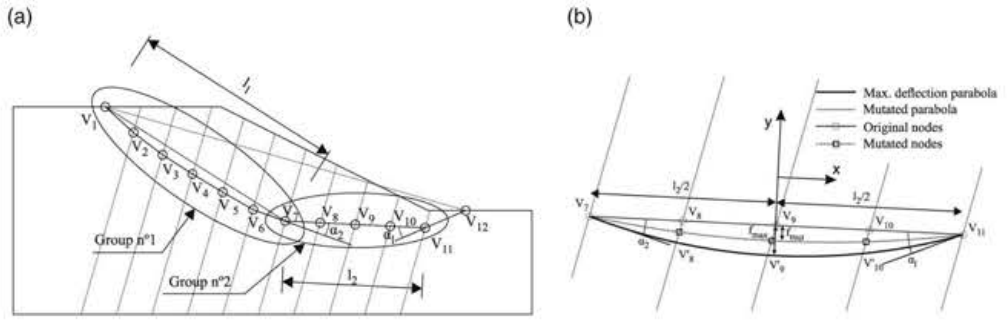


Figure 6. Description of the parabolic mutation operator.

(2) then, a 'flag' is sampled from a Bernoulli distribution; depending on such flag, V_i is translated, along its 'allowable extreme interval', in the direction of P_{Li} (*i.e.* over $l_{i,1}$) or of P_{Ui} (*i.e.* over $l_{i,2}$);
(3) the translation distance, δ_i, over $l_{i,1}$ or $l_{i,2}$, is given by $\delta_i = l_{i,j} f(g)$, where j is the translation direction ($j = 1$ or $j = 2$), and $f(g)$ is the 'damping function' employed in the non-uniform mutation. This generates a mutated node, V_i';
(4) using the new positions of both extreme nodes, V_1' and V_N', the remaining (interior) nodes are obtained using steps 2 to 5 of the procedure to generate individuals of the initial population (Section 2.3).

3.2.5. *Parabolic mutation*

The parabolic mutation operator initially identifies groups of 'aligned' nodes. Three consecutive vertices, V_{i-1}, V_i and V_{i+1}, are 'aligned' when the acute angle formed by $V_{i-1}V_i$ and V_iV_{i+1} is smaller than a reference angle, β. (β can be random, as sampled from a $[0, \beta_{\max}]$ uniform distribution, in which $\beta_{\max}$ is user-defined.) Figure 6 shows an example with two groups of 'aligned' nodes.

Then, one group of aligned nodes is randomly selected (with equal selection probability), to mutate its nodes. Let's imagine that Group 2 was selected. The parabolic mutation operator first defines the segment, l_2, that joins its initial and final nodes (V_7 and V_{11} in the example); next, the angles, α_1 and α_2, between segments adjacent to the group of aligned nodes (in the example, $V_{11}V_{12}$ and V_6V_7) with l_2 are computed.

Using α_1 and α_2, an auxiliary 'maximum deflection parabola' is considered with the following constraints (see Figure 6(b)):

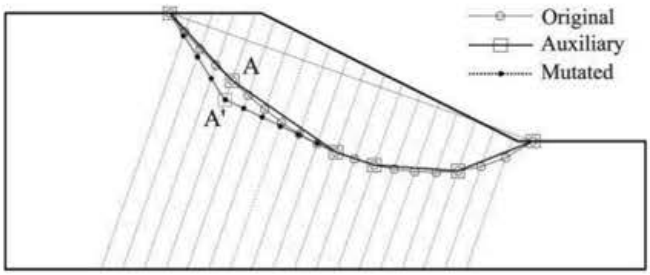


Figure 7. Illustration of the 'random lumped mutation' operator.

(i) it passes through both extremes of the aligned group,
(ii) its axis of symmetry is perpendicular to l_2 and passes through its midpoint, and
(iii) its angle with l_2 at the extreme nodes is equal to the minimum of α_1 and α_2 (or of the single available value of α, if the group of aligned nodes starts or finishes at the slope boundary).

If nodes lie along the 'maximum deflection parabola', the resulting polygonal would strictly fulfil upward-concavity. But, for greater flexibility, the 'maximum deflection parabola' is only a reference, and the 'deflection' of the 'mutated parabola' is $f_{\text{mut}} = \xi f_{\text{max}}$. ($f_{\text{max}}$ is the distance between the vertex of the 'maximum deflection parabola' and l_2.) Using f_{mut}, the mutated parabola results from the condition that it passes through three points: the two end nodes and the new vertex with f_{mut} deflection.

The nodes of the mutated slip surface can then be computed as the intersection between the 'mutated parabola' and the auxiliary perpendicular lines of the original slip surface (see Figure 6(b)).

3.2.6. *Random lumped mutation*

This operator, designed for slip surfaces with many nodes, aims to align and displace several nodes of the slip surface. First, N_k out of the $N - 2$ interior nodes within the slip surface polygonal are randomly chosen ($N_k = 4$ are used herein); using such nodes, in addition to the original polygonal's extreme nodes, an auxiliary polygonal, with $N_k + 2$ nodes, and that shares its nodes with the original slip surface, is produced (see Figure 7).

Next, one (randomly selected, interior or exterior) node from the auxiliary slip surface is mutated. This mutation, which is randomly chosen to be uniform or non-uniform (see Section 3.2.1), displaces one node of the auxiliary slip surface (*e.g.* from A to A' in Figure 7). The mutated slip surface—again with N nodes—will result as the union of:

(i) the two straight segments, with their interior nodes included, that result from 'recoding'—*i.e.* from intersecting with the auxiliary perpendicular lines, L; see Section 2.5—the two segments of the auxiliary slip surface connected with the mutated node, A'; and
(ii) the original slip surface for the remaining nodes. (Of course, if the mutated node is one of the extreme nodes, a 'recoding' of the whole slip surface is also needed at the end.)

3.2.7. *Straight lumped mutation*

This operator is similar to the 'random lumped mutation' operator, with the difference that nodes of the 'auxiliary' slip surface correspond to the extremes of groups of 'aligned' nodes (see Section 3.2.5). That is, groups of 'aligned' nodes are initially selected; next, their extreme nodes are joined with the extreme nodes of the slip surface (in cases where they had not already

been selected). These nodes constitute the 'auxiliary' slip surface to which the random lumped mutation (see Section 3.2.6) is applied.

4. Efficiency of genetic algorithm operators

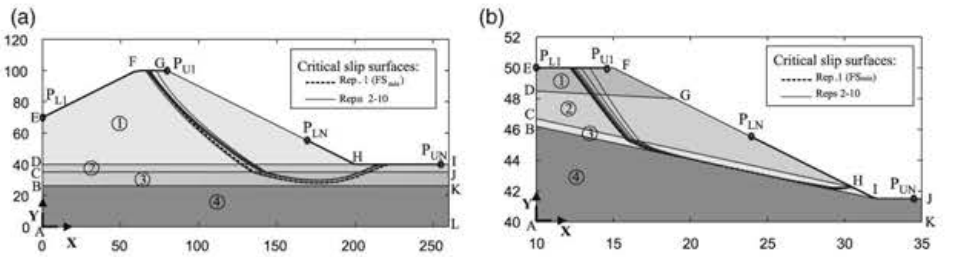
Three 'efficiency indices' (P1 to P3) are developed to analyse the GA operators presented above. They are as follows.

(1) 'The percentage of offspring that improve the previous solution' (P_1): for a given operator and group of generations, P_1 is obtained by dividing the number of offspring (generated by that operator in that group of generations) that improve the (overall) best FS of their parents' generation, by the total number of offspring generated by that operator in that group of generations.
(2) 'The percentage of offspring that improve on their parent(s)' (P_2): for a given operator and group of generations, the number of offspring (generated by that operator in that group of generations) that improve on their parents—and not the previous generation as a whole—is divided by the total number of offspring generated by that operator in that group of generations.
(3) 'The percentage relative reduction of FS' (P_3): the absolute reduction of (minimum) FS between the first and last generations, $\mathrm{RFS} = \mathrm{FS}_{\min}(1) - \mathrm{FS}_{\min}(N_{\mathrm{gen}})$, is computed first. Then, for a given operator and generation, the relative reduction of FS (with respect to RFS, and compared with the minimum FS for the previous generation) due to the best offspring produced by that operator in that generation is computed, making it equal to zero if the result is negative—*i.e.* if the operator did not improve on the previous generation. To compute P_3 for a group of generations, for all generations within the group, the percentage reductions computed in such a way are summed.

5. Application examples

Next, application examples are presented. The first two examples are traditional benchmarks from the literature; the third is a new one inspired by a real project. For each example, between 10 and 12 runs, or 'replications', using 250 generations of populations with 100 individuals, are employed. To create a new generation, each of the two crossover operators and eight mutation operators described in Section 3 is applied five times, hence producing a total of $N_{\mathrm{off}} = 55$ new offspring per generation. (The arithmetic crossover operator produces two offspring; the others produce one. Note also that in Section 3.2.1 two mutation operators are explained simultaneously.) The selection of individuals is rank-based (see Section 2.4) and uses elitism: the best $N_{\mathrm{elit}} = 5$ individuals always pass to the next generation. In this work, $q = 0.01$ is used for the first $2/3$ out of the $N_{\mathrm{gen}} = 250$ generations considered, and $q = 0.1$ for the remaining. (This is to select individuals for both reproduction and killing.) With such a selection of q values, 'exploration' is encouraged at initial GA stages, whereas 'refinement'—also called 'exploitation'—is sought at later stages.

To analyse the influence of the number of nodes, in Examples 1 and 2 trial slip surfaces with $N = 13$ and $N = 20$ nodes are tested; however, since the GA provides very similar results in both cases, for Example 3 only slip surfaces with $N = 13$ nodes are used. For Examples 1 and 2, the entry (active) and exit (passive) angles of the slip surface—measured with respect to the X-axis and positive clockwise—are limited to be, respectively, within [45, 60] and [135, 200] intervals; changing these limits slightly does not influence the results. For Example 3, the range of accepted entry (active) angles is the same, whereas, following Jumiki's considerations presented in Duncan and Wright (2005), a [135, 163] interval is used for the range of accepted exit (passive) angles; in



Layer	c' (kPa)	ϕ' (degrees)	γ (kN/m³)
1	49.00	29	20.38
2	0.00	30	17.64
3	7.84	20	20.38
4	0.00	30	17.64

	X	Y		X	Y
A	0	0	G	80	100
B	0	26	H	200	40
C	0	35	I	260	40
D	0	40	J	260	35
E	0	70	K	260	26
F	60	100	L	260	0

Layer	c' (kPa)	ϕ' (degrees)	γ (kN/m³)
1	15.0	20	19.0
2	17.0	21	19.0
3	5.0	10	19.0
4	35.0	28	19.0

	X	Y		X	Y
A	10	40	G	19	48
B	10	46.2	H	30.5	42.25
C	10	46.7	I	32	41.5
D	10	48.5	J	35	41.5
E	10	50	K	35	40
F	15	50			

Figure 8. Examples 1(a) and 2(b): geometry and geotechnical properties.

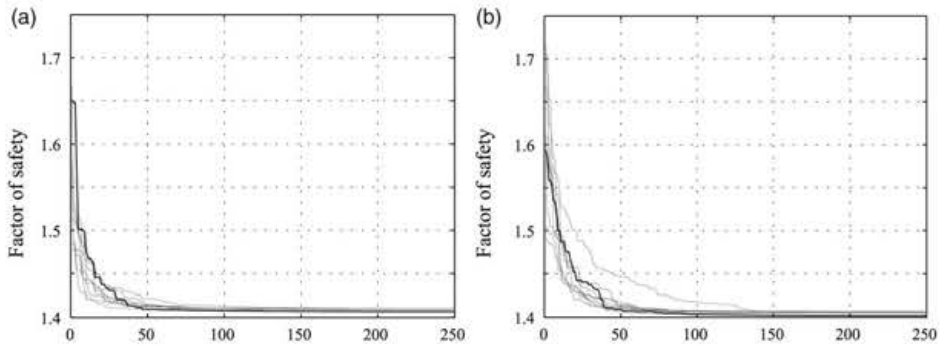


Figure 9. Evolution of FS with the number of generations for replications in Example 1. (a) $N = 13$; (b) $N = 20$.

this case, changing this interval affects the results, and slightly different minimum FS values are obtained if these limits are relaxed.

5.1. *Examples 1 and 2*

Example 1 analyses a slope with horizontal layers that was previously employed by Greco (1996, example 4), referring to Yamagami and Ueta (1988). Example 2 analyses the slope 'with complex soil layering' considered in Zolfaghari, Heath, and McCombie (2005, figure 13). Geometrical and geotechnical details are shown in Figure 8. Figure 8 also shows the slip surfaces with minimum FS values for replications with $N = 20$ nodes. (Slip surfaces for $N = 13$ are similar and not presented herein. Videos showing convergence with the number of generations, and a table with the coordinates of nodes of the minimum FS slip surfaces, are provided as 'supplemental data' online—see the unnumbered section before the references list).

Figure 9 shows the evolution, with the number of generations, of the minimum FS for the 10 replications considered in Example 1. (The evolution for Example 2 is similar and not presented herein.)

Table 1 lists computed final (minimum) FS values for each replica.

Using box-plots to represent the results of all replications, Figure 10 shows the overall computed efficiency indices (Section 4) for one single group of 250 generations. In addition, since efficiency can change during convergence—for instance, due to different capabilities for (early) 'exploration'

Table 1. Minimum FS solutions for replications in Examples 1–3.

	Example 1		Example 2		Example 3
Replica	$N = 13$	$N = 20$	$N = 13$	$N = 20$	$N = 13$
1	1.406814	1.401968	1.105457	1.106906	0.968952
2	1.407340	1.402143	1.106261	1.107459	0.969213
3	1.407413	1.405359	1.106300	1.108110	0.969528
4	1.407752	1.405401	1.106431	1.108277	0.969614
5	1.409481	1.405414	1.106912	1.109383	0.970011
6	1.409484	1.405611	1.107216	1.109620	0.970275
7	1.409505	1.405838	1.108424	1.114229	0.971280
8	1.409549	1.405919	1.109896	1.118895	0.971309
9	1.410333	1.406245	1.110103	1.119569	0.971509
10	1.410377	1.407049	1.111116	1.124739	0.992334
11	–	–	1.111796	–	–
12	–	–	1.114197	–	–
Max. diff.	0.035	0.005	0.00874	0.017833	0.023382

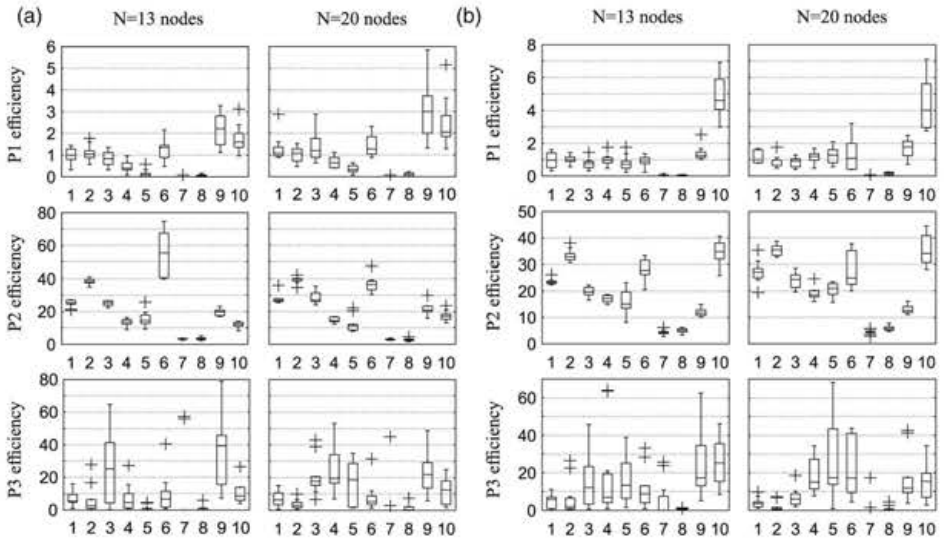


Figure 10. Overall efficiency of genetic algorithm operators for (a) Example 1 and (b) Example 2. Operators Key: 1 = uniform mutation; 2 = non-uniform mutation; 3 = total non-uniform mutation; 4 = random lumped mutation; 5 = straight lumped mutation; 6 = parabolic mutation; 7 = extreme vertices mutation; 8 = straight mutation; 9 = arithmetic crossover; 10 = heuristic crossover.

and for (later) 'refinement'—Figure 11 presents its evolution, for the case of $N = 20$ nodes, when 5 groups of 50 generations are considered.

5.2. *Example 3*

Finally, one 'mechanical stabilized earth' (MSE) bridge abutment is considered. An excavation supported with a gravity wall was considered next to it, with the objective of constructing an underpass below one auxiliary road underneath the bridge. Figure 12(a) presents the cross-section of the bridge abutment, with the proposed excavation and gravity wall in place. It also shows the geotechnical properties considered, as well as some representative design loads. Figure 12(b) presents a general view. The goal of the analysis—which is presented herein in a simplified

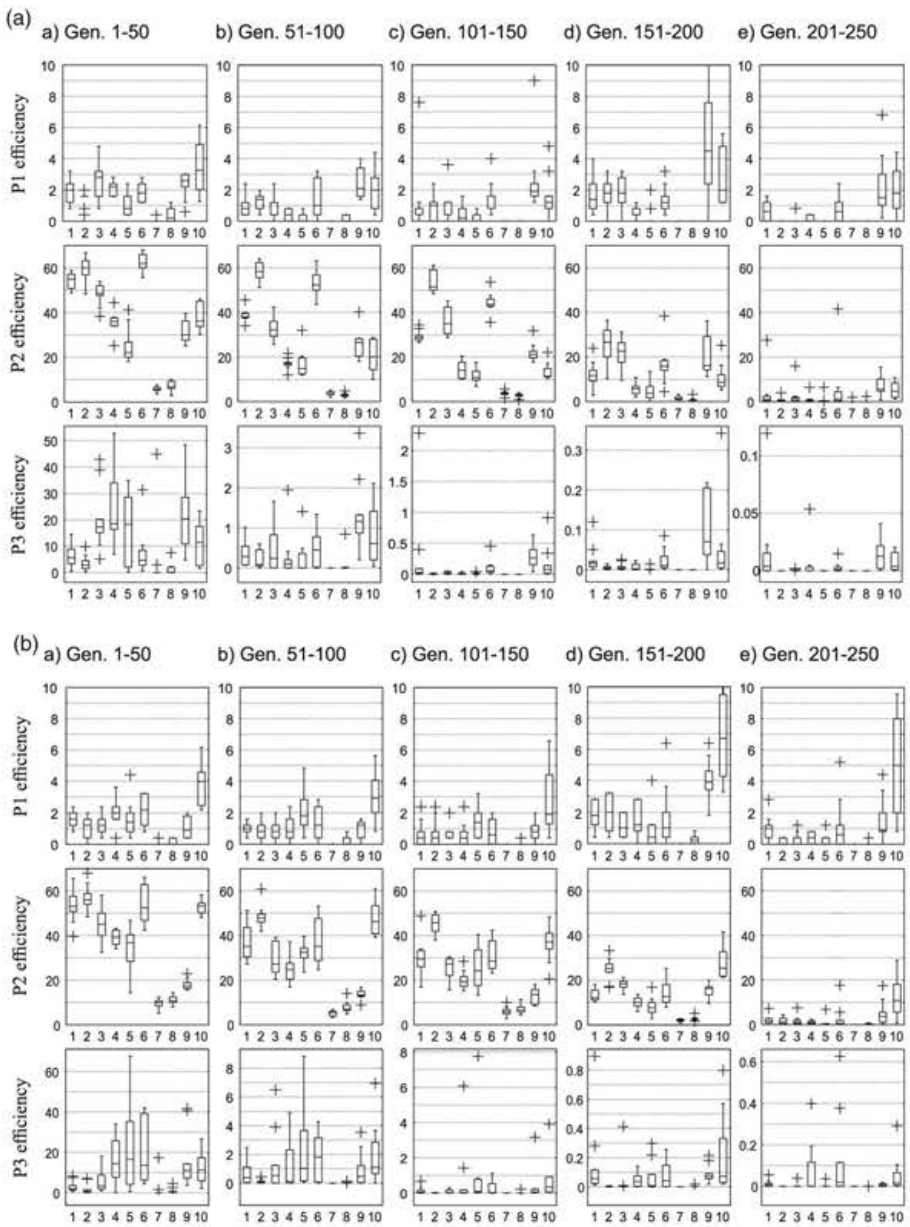
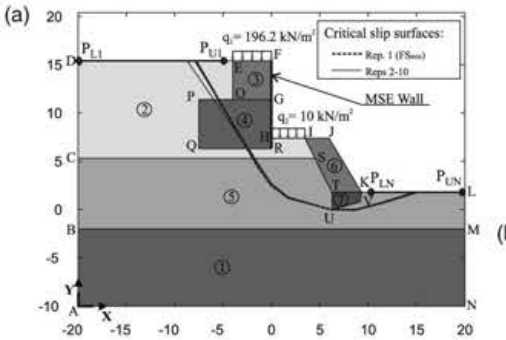


Figure 11. Evolution of efficiency of genetic algorithm operators, for groups of 50 generations during convergence of Examples 1 and 2 with $N = 20$ nodes. Operators Key: as for Figure 10.

and not necessarily representative form—is to identify the *deep* slip surface with minimum FS. That is, toe surfaces—which might be critical numerically, although perhaps should be better analysed using MSE design procedures—or surfaces through the retaining wall, are not considered.

Table 1 lists the final (minimum) FS values computed with each replica. Coordinates of the slip surface with minimum FS are listed as 'supplemental data' online (see the unnumbered section before the references list).



Layer properties

Layer	c' (kPa)	φ' (degrees)	γ (kN/m³)
1	10.0	35	23.0
2	5.0	30	20.0
3	500.0	0	23.0
4	50.0	36	24.0
5	10.0	25	20.0
6	0.0	40	19.0
7	500.0	0	23.0
8	0.0	0	10.0

Point coordinates (m)

	X	Y
A	-20	-10
B	-20	-2
C	-20	5.3
D	-20	15.4
E	-4	15.4
F	0	15.4
G	0	11.4
H	0	7.4
I	3.45	7.4
J	5.95	7.4
K	9.3	1.8
L	20	1.8
M	20	-2
N	20	-10
O	-4	11.4
P	-7.5	11.4
Q	-7.5	6.3
R	0	6.3
S	4.5	5.3
T	6.25	1.8
U	6.3	0
V	9.2	0.8

Figure 12. Example 3: (a) geometry and geotechnical properties; (b) overview of excavation.

6. Discussion of results

6.1. *Reliability, flexibility, robustness and convergence*

The results in Section 5 illustrate that the proposed GA finds very good solutions. For reference, Tables 2 and 3 compile minimum FS values reported in the literature for Examples 1 and 2.

Results also show that—although solutions are sometimes not the best up to the last decimal digit—the proposed GA's performance is, at least, comparable to the other methods considered. However, the results in Tables 2 and 3 should only be taken as approximate performance indicators. The reason is that the computer codes employed are different: even using the same limit equilibrium method, differences could arise from the location of the resultant at the base, or from convergence tolerances. Also, some of the best solutions reported in the literature may correspond to non-physical solutions, in which the $P_{\alpha,i} = \cos(\alpha_i - \theta)[1 + \tan\phi_i \tan(\alpha_i - \theta)/\mathrm{FS}]$ denominator that appears in the formulation of Spencer's method—see for example Duncan and Wright (2005)—falls below the 0.3–0.4 minimum threshold recommended by Baker (1980) so that 'the Spencer method converges and yields reasonable results' (Malkawi, Hassan, and Sarma 2001).

In any case, it seems clear that the proposed GA significantly improves the simple GA algorithm (Zolfaghari, Heath, and McCombie 2005) and methods based on Tabu and ant-colony optimization, which sometimes fail to find optimum solutions. It also improves the solutions provided by simulated annealing (SA), by chaos optimization, by simple and modified harmony search (SHS and MHS), by some other GAs without specifically designed custom operators (Cheng, Li, and Chi 2007), as well as by other traditional approaches such as the BFGS and DFP methods (see Tables 2 and 3).

Given that the minimum FS results are similar, other aspects—such as 'flexibility', 'convergence speed' and 'robustness'—can also be considered to compare the proposed GA with the remaining methods (MC, PSO and MPSO, SBGA, AFSA, RCGA, HM/PSO and ABC).

To assess 'flexibility', the influence of the number of nodes, N, on the computed minimum FS solutions is studied. As shown in Tables 2 and 3, the proposed GA finds very similar results for

Table 2. Comparison of minimum FS values reported in the literature for Example 1 (for Spencer's method, unless reported otherwise).

Source	Search method[a]	Minimum FS
Yamagami and Ueta (1988)	Several	1.423, 1.453, 1.402, 1.405[b]
Greco (1996)	MC	1.400–1.406, 1.400–1.413[c]
Malkawi, Hassan, and Sarma (2001)	MC	1.333[d]
Sarma and Tan (2006)	Crit. accel.	1.422[e]
Cheng *et al.* (2007)	PSO	1.3862, 1.3942, 1.3983[f]
Cheng *et al.* (2007)	Modif. PSO	1.4017, 1.3967, 1.4046
Sun, Li, and Liu (2008)	SBGA	1.395[g]
Kahatadeniya, Nanakorn, and Neaupane (2009)	ACO	1.348[h]
Kang, Li, and Ma (2013)	ABC	1.3776, 1.3785[i]
Hu *et al.* (2013)	Chaos optim.	1.4142[j]
This work	Custom RCGA	1.4068–1.4104, 1.4020–1.4070[k]

[a] MC = Monte Carlo; GA = Genetic algorithm; PSO = Particle swarm optimization; ACO = Ant Colony Optimization; SBGA = Spline-based GA; RCGA = Real-coded GA; ABC = Artificial Bee Colony.
[b] For BFGS, DFP, Powell and Simplex methods, respectively.
[c] For $N = 13$ nodes; pattern search and random walk, respectively.
[d] Cheng *et al.* (2007) report FS = 1.39 for their best slip surface.
[e] Using Sarma's method.
[f] For $N = \{15, 20, 30\}$ nodes, respectively.
[g] Solution defined by four nodal points connected by splines. Using SLOPE/W, FS $\approx$ 1.43 was obtained for their failure surface (approximated; as scanned from their figures).
[h] Probably a non-physical solution. Using the commercial software SLOPE/W, FS $\approx$ 1.46 was obtained for their failure surface (approximated; as scanned from their figures).
[i] For $N = 31$ nodes. Best of 30 experiments, for two values of the population size parameter. (NP = 20 and NP = 50.)
[j] For $N = 15$.
[k] Ranges of values for 10 replications and for $N = \{13, 20\}$ nodes, respectively (see Table 1).

both $N = 13$ and $N = 20$, with only a 0.36% difference between the best results computed for Example 1 and a 0.13% difference for Example 2. (This explains why only slip surfaces with $N = 13$ nodes are used for Example 3.) These differences compare well with other methods: of those methods providing better predictions that report the influence of N, only the 'particle swarm optimization' (PSO) and 'modified PSO' (MPSO) methods (Cheng *et al.* 2007; Cheng, Li, and Chi 2007), as well as the 'Real-coded GA' (RCGA) (Li *et al.* 2010), have a similar capability to find good solutions independently of N. Kang, Li, and Ma (2013) do not report the influence of N on the performance of the Artificial Bee Colony algorithm (all results correspond to $N = 31$), although results seem to depend on the population size.

To assess 'robustness', the capability to provide similar results in different replications is considered. Results show that, after conducting 10–12 replications with the proposed GA algorithm, only (small) FS differences of about $\{0.3, 0.4, 0.8, 1.6\}$% occur, in Examples 1 and 2, and with $N = 13$ and $N = 20$. This is significantly smaller than the 5.6% difference reported for 9 analyses with the AFSA method. Although it is comparable to the 0.45% reported by the RCGA, the RCGA result corresponds to only 3 replications instead of 10–12. Although Kang, Li, and Ma (2013) do not report all the FS results for their 30 experiments, the Artificial Bee Colony algorithm is likely to be (at least for this example) less robust than the proposed GA. For instance, for Example 1 (Example 3 in their paper), they report an average FS that is clearly above 1.2; since their minimum FS is below 1.1, there must be either some very bad results or many experiments with FS above 1.2–1.3. In Example 3, the difference in FS for the ten replications considered is about 2.4% or, in absolute FS value, 0.023.

Finally, as expected, results show that convergence is faster with fewer nodes. Figure 9, and the observation of convergence for the other examples, illustrates that convergence is fast, with only about 50–100 generations needed to get good FS estimations with $N = 13$ nodes and 100–150 generations needed with $N = 20$. Given that the population size is 100, and that 55 new offspring are generated for each generation, this represents about 5545–8295 tested slip surfaces. (It must

Table 3. Comparison of minimum FS reported in the literature for Example 2 (for Spencer's method, unless reported otherwise).

Source	Search method[a]	Minimum FS
Zolfaghari, Heath, and McCombie (2005)	Simple GA	1.24[b]
Sarma and Tan (2006)	Crit. accel.	1.091[c]
Cheng (2007)	SA	1.12[d]
Cheng *et al.* (2007)	PSO	1.1055, 1.1095, 1.1010[e]
Cheng *et al.* (2007)	Modif. PSO	1.1139, 1.1174, 1.1289
Cheng, Li, and Chi (2007)	SA	1.1789,1.2564,1.2813[f]
Cheng *et al.* (2007)	GA	1.1495,1.1117,1.1440
Cheng *et al.* (2007)	PSO	1.1080,1.1341,1.1095
Cheng *et al.* (2007)	SHM	1.2512,1.2405,1.2068
Cheng *et al.* (2007)	MHM	1.1509,1.1315,1.1385
Cheng *et al.* (2007)	Tabu	1.4714,1.4661,1.4650
Cheng *et al.* (2007)	ACO	1.4249,1.5299,1.5817
Cheng *et al.* (2008b)	AFSA	1.1195–1.1828[g]
Kahatadeniya, Nanakorn, and Neaupane (2009)	ACO	1.361
Li *et al.* (2010)	RCGA	1.113,1.115,1.117[h]
Li *et al.* (2010)	RCGA	1.114,1.116,1.119
Cheng *et al.* (2012)	HM/PSO	1.09[i]
Rickard and Sitar (2012)	CoDE	1.12[j]
Kang, Li, and Ma (2013)	ABC	1.0857, 1.1282[k]
Hu *et al.* (2013)	Chaos optim.	1.1471[l]
This work	Custom RCGA	1.1055-1.1142,1.1069-1.1247[m]

[a]Key: GA = Genetic algorithm; PSO = Particle swarm optimization; SA = Simulated annealing; SHM = Simple harmony search; MHM = Modified harmony search; ACO = Ant Colony Optimization; AFSA = Artificial fish swarms algorithm; RCGA = Real-coded GA; CoDE = Composite differential evolution; ABC = Artificial Bee Colony.
[b]Using the Morgenstern–Price method.
[c]Using Sarma's method. In addition to the different method for stability analysis, an assumed tension crack 'might be the primary reason for these lower values of FS_{min}' (Li *et al.* 2010).
[d]Method unspecified. Results for $N = 6$ nodes.
[e]Results for $N = \{15, 20, 30\}$ nodes, respectively.
[f]Results for $N = \{21, 31, 41\}$ nodes, respectively.
[g]The authors report values between 1.1195 and 1.1828 for 9 different cases considered.
[h]Morgenstern–Price method. Results for three analyses with different random seeds.
[i]No information provided about flexibility, robustness or convergence rate.
[j]Based on M. Tabarroki's code.
[k]For $N = 31$ nodes. Best of 30 experiments, for two values of the population size parameter. ($NP = 20$ and $NP = 50$).
[l]For $N = 15$.
[m]Ranges of values for 10–12 replications and for $N = \{13, 20\}$ nodes, respectively (see Table 1).

be emphasized that this rate of convergence is achieved while maintaining the robustness of the GA, hence avoiding premature convergence to a local optimum.)

This is comparable to the orders of magnitude employed by Malkawi, Hassan, and Sarma (2001) (up to 10,000), and it is also similar to the number of evaluations reported by Kang, Li, and Ma (2013) (between 3320 and 10,000, depending on population size). It is also slightly larger than the range of values (up to 1000) reported by Greco (1996) for a different example (the number of surfaces tested for Example 1 is not reported). However, they are lower than the ranges presented by Cheng *et al.* (2008a) for their harmony search methods: depending on the generation method (P2, P3 or P4), they report approximately 10,000–25,000 slip surfaces to solve Example 2 with $N = 21$. (Although their P5 generation method needs only 3881 tested surfaces, they do not recommend it given 'the limitations associated with P5 for complicated problems'—Cheng *et al.* [2008a].) They are also lower than the number of slip surfaces reported, for Example 2, by Cheng *et al.* (2007) for the MPSO method (22,146 for $N = 21$) and significantly lower than for the PSO method (92,950 for $N = 21$).

Other methods considered by Cheng, Li, and Chi (2007) also require a higher number of slip surface evaluations for $N = 21$: 53,104, 33,584, 18,270, 12,815, 32,388, 748 and 5913 are required to achieve minimum FS solutions for, respectively, the SA, GA, PSO, SHM, MHM,

Tabu and ACO methods. (Remember that Tabu and ACO provide inadequate FS estimates.) Sun, Li, and Liu (2008) show that their spline-based algorithm is very efficient, since they only need 600 evaluations to locate the four nodes that define their critical slip surface in Example 1. (However, their reported slip surface is probably too deep, with an FS that seems to correspond with a non-physical solution associated to a negative P_α.) By using 'dynamic vertex addition', Li *et al.* (2010) also report an efficient convergence, with only 3320 FS evaluations needed to solve Example 2 with $N = 31$, whereas Rickard and Sitar (2012) report that '8000 slip surfaces were considered [...] though convergence generally occurred closer to 4000 iterations'. Finally, although the number of FS evaluations conducted by Cheng *et al.* (2012) to achieve their FS = 1.09 for Example 2 is not reported, it is possibly higher than 40,000, since they report 42,308 evaluations for another example with $N = 21$, which is 'less difficult in terms of optimization' (Cheng *et al.* 2012).

6.2. *Efficiency of GA operators*

The results show that crossover operators tend to have (overall) high P_1 and P_2 efficiencies, hence indicating a significant contribution to improving on their parents and the existing best solutions. Arithmetic crossover has higher P_1 and P_2 values than heuristic crossover for problems with curved/non-confined slip surfaces (as in Example 1); and lower values when the slip surface is constrained to a thin layer (as in Example 2). Similarly, P_3 efficiencies show that crossover operators are responsible for much of the FS reduction, specially when the number of nodes is lower (*i.e.* when $N = 13$). Furthermore, the (relative) efficiency of crossover operators, with respect to mutation operators, is maintained during convergence.

Mutation operators also contribute—sometimes significantly—to reducing FS. Although they often improve on their parents (they have a similar P_2 to crossover), they are not (overall) as efficient at improving the best available solution (hence, they have a generally lower P_1). Also, note that, despite their somewhat chaotic behaviour—mutations are inherently random in nature—they are crucial to reduce P_3 during both initial 'exploration' and later 'refinement'.

In addition, the efficiency of mutation operators depends on the type of critical slip surface—non-confined versus confined; or mainly curved versus mainly straight—and on its number of nodes. For example, the mutation of single nodes sometimes helps to reduce FS significantly, specially for lower N values (*i.e.* for $N = 13$); the uniform mutation maintains its (relative) improvement capabilities during convergence, but the non-uniform mutation produces almost negligible FS improvements in later generations. The total non-uniform operator—which produces an offspring that, in early generations, can be very different from its parent, therefore helping in the 'exploration' of the search domain—works better for lower N and for early generations, producing only minor refinements in later generations. In addition, it works better for 'mainly-curved' optimum slip surfaces (as in Example 1) than for 'mainly-straight' surfaces (as in Example 2). Similarly, the parabolic mutation operator is efficient for 'mainly-curved' surfaces (Example 1), and even better for 'mainly-straight' surfaces with many nodes (Example 2). On the other hand, given their design, the random and straight lumped mutation operators work better when N is higher (for $N = 20$), and they significantly contribute to reducing FS, specially for constrained slip surfaces (Example 2). They also maintain their (relative) improvement capabilities during all stages of convergence, therefore becoming crucial for the GA's success.

The extreme vertices mutation operator and the straight mutation operator (which allows the slip surface to be located within existing soft, and relatively thin, strata) also contribute, albeit very slightly, to reducing the FS. However, they are included because they may 'couple' with other operators: *i.e.* producing offspring that are further improved later by another operator.

7. Conclusions

Heuristic optimization methods are a promising tool for slope stability analyses based on limit equilibrium, since they avoid problems with local minima that affect other (deterministic) optimization methods. Despite initial efforts to use genetic algorithms (GAs) for this geotechnical problem, some initial performance problems caused subsequent research to focus mainly on alternative approaches. However, in agreement with recent research (Sun, Li, and Liu 2008; Li *et al.* 2010; Rickard and Sitar 2012), this work demonstrates that GAs can identify critical slip surfaces both robustly and efficiently.

A new GA is proposed that has a standard structure but with a novel encoding and generation of individuals, and with custom-designed operators for mutation and crossover. In particular, eight new mutation operators—uniform, non-uniform, total non-uniform, random lumped and straight lumped, parabolic, extreme vertices, and straight mutations—plus two crossover operators—arithmetic and heuristic crossover—are defined and employed.

Such custom-built GA encoding and operators have been specifically designed for the slope stability problem considered, so that

(i) they only produce kinematically feasible slip surfaces during generation of the initial population; and
(ii) they produce feasible slip surfaces during crossover and mutation with a high probability, hence easing the optimization and increasing the convergence rate.

The performance of the proposed GA—predictive capability, flexibility, robustness and convergence—is analysed using two typical benchmark examples from the literature and one new example inspired by practical experience. The results indicate that the proposed GA presents adequate predictive capabilities, providing FS results that either improve or are very similar to other minimum FS solutions reported in the literature. Similarly, it is flexible—providing similar results when N changes—and robust—providing similar results for different replications. This is considered an important aspect of the proposed GA, since it allows designers to trust their results without the need to conduct a large number of replications.

Furthermore, the algorithm is efficient, since a relatively low number of FS evaluations is needed to minimize the FS. Although the proposed GA is not designed to be the fastest, it combines convergence speed and robustness; however, in any case, putting too much emphasis on rapid convergence should be avoided. It is well known—see for example Michalewicz (1996)—that algorithms designed for fast convergence may reduce the 'exploratory' capabilities of the GA, therefore increasing the likelihood that the solution becomes trapped in local minima.

Finally, new indices for expressing the efficiency of GA operators are developed and presented. (They analyse the operators' potential to improve on the offspring's parents or the overall previous generation, and to reduce the minimum FS value during convergence towards the minimum.) They show that crossover operators greatly contribute to reducing FS values during all stages of convergence; they are therefore helpful during both 'exploration' and 'exploitation' stages. Lumped mutation operators—random lumped mutation and straight lumped mutation—have a similarly important contribution towards GA success, specially for slip surfaces with a larger number of nodes. They help to increase the GA's robustness, as they continue to explore the search domain during all generations; also, as they generate straight segments, they combine well with the parabolic mutation operator, which curves such segments in subsequent generations. Other mutation operators, however, have somewhat reduced efficiencies that are further affected by the shape of the critical slip surface or by its number of nodes.

These observations, which are new in the GA literature on slope stability, could lead to 'customized' GAs for improved slope stability analyses—which, for instance, would employ operators with a frequency that depends on the type of problem considered, or on the generation number.

References

Baker, R. 1980. "Determination of the Critical Slip Surface in Slope Stability Computations." *International Journal of Numerical and Analytical Methods in Geomechanics* 4 (4): 333–359.

Bolton, H. P. J., G. Heymann, and A. A. Groenwold. 2003. "Global search for critical failure surface in slope stability analysis." *Engineering Optimization* 35 (1): 51–65.

Chen, Zu-Yu. 1992. "Random Trials Used in Determining Global Minimum Factor of Safety of Slopes." *Canadian Geotechnical Journal* 29 (2): 225–233. doi:10.1139/t92-026.

Cheng, Y. M. 2003. "Location of Critical Failure Surface and Some Further Studies on Slope Stability Analysis." *Computers and Geotechnics* 30 (3): 255–267.

Cheng, Y. M. 2007. "Global Optimization Analysis of Slope Stability By Simulated Annealing With Dynamic Bounds and Dirac Function." *Engineering Optimization* 39 (1): 17–32.

Cheng, Y. M., L. Li, and S. C. Chi. 2007. "Performance Studies on Six Heuristic Global Optimization Methods in the Location of Critical Slip Surfaces." *Computers and Geotechnics* 34 (6): 462–484.

Cheng, Y. M., L. Li, S.-C. Chi, and W. B. Wei. 2007. "Particle Swarm Optimization Algorithm for the Location of the Critical Non-Circular Failure Surface in Two-Dimensional Slope Stability Analysis." *Computers and Geotechnics* 34 (2): 92–103.

Cheng, Y. M., L. Li, T. Lansivaara, S. C. Chi, and Y. J. Sun. 2008(a). "An Improved Harmony Search Minimization Algorithm Using Different Slip Surface Generation Methods for Slope Stability Analysis." *Engineering Optimization* 40 (2): 95–115.

Cheng, Y. M., L. Li, Y. J. Sun, and S. K. Au. 2012. "A Coupled Particle Swarm and Harmony Search Optimization Algorithm for Difficult Geotechnical Problems." *Structural and Multidisciplinary Optimization* 45 (4): 489–501.

Cheng, Y. M., L. Liang, S. C. Chi, and W. B. Wei. 2008(b). "Determination of the Critical Slip Surface Using Artificial Fish Swarms Algorithm." *Journal of Geotechnical and Geoenvironmental Engineering (ASCE)* 134 (2): 244–251.

Duncan, J. M., and K. S. Song. 1984. *SLOPE8R: A Computer Program for Slope Stability Analysis With Non-Circular Slip Surfaces*. Technical Report, Department of Civil Engineering, University of California, Berkeley, CA.

Duncan, J. M., and S. G. Wright. 2005. *Soil Strength and Slope Stability*. Hoboken, NJ: Wiley.

Goldberg, D. E. 1989. *Genetic Algorithm in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley.

Greco, V. R. 1996. "Efficient Monte Carlo Technique for Locating Critical Slip Surface." *Journal of Geotechnical Engineering* 122 (7): 517–525.

Hu, C., R. Jimenez, S. Li, and L. Li. 2013. "Determination of Critical Slip Surfaces Using Mutative Scale Chaos Optimization." *Journal of Computing in Civil Engineering*, December 4. doi:10.1061/(ASCE)CP.1943-5487.0000373.

Jong, J.-C. 1998. "Optimizing Highway Alignments with Genetic Algorithms." PhD diss. University of Maryland, College Park.

Kahatadeniya, K. S., P. Nanakorn, and K. M. Neaupane. 2009. "Determination of the Critical Failure Surface for Slope Stability Analysis Using Ant Colony Optimization." *Engineering Geology* 108 (1–2): 133–141.

Kang, F., J. Li, and Z. Ma. 2013. "An Artificial Bee Colony Algorithm for Locating the Critical Slip Surface in Slope Stability Analyis." *Engineering Optimization* 45 (2): 207–223.

Li, Y.-C., Y.-M. Chen, T.-L. T. Zhan, D.-S. Ling, and P. J. Cleall. 2010. "An Efficient Approach for Locating the Critical Slip Surface in Slope Stability Analyses Using a Real-Coded Genetic Algorithm." *Canadian Geotechnical Journal* 47 (7): 806–820.

Malkawi, A. I. H., W. F. Hassan, and S. K. Sarma. 2001. "Global Search Method for Locating General Slip Surface Using Monte Carlo Techniques." *Journal of Geotechnical and Geoenvironmental Engineering* 127 (8): 688–698.

McCombie, P., and P. Wilkinson. 2002. "The Use of the Simple Genetic Algorithm in Finding the Critical Factor of Safety in Slope Stability Analysis." *Computers and Geotechnics* 29 (8): 699–714.

Michalewicz, Z. 1996. *Genetic Algorithms + DataStructures = Evolution Program*. 3rd ed. New York: Springer.

Mitchell, M. 1996. *An Introduction to Genetic Algorithms*. Cambridge, MA: MIT Press.

Rickard, O. C., and N. Sitar. 2012. *bSLOPE: A General Limit Equilibrium Slope Stability Analysis Code for iOS*. Geotechnical Engineering Report No. UCB/GT/12-01, Department of Civil and Environmental Engineering, University of California, Berkeley, CA.

Sarma, S. K., and D. Tan. 2006. "Determination of Critical Slip Surface in Slope Analysis." *Geotechnique* 56 (8): 539–550.

Schuster, R. L. 1996. "Socieconomic Significance of Landslides." In *Landslides: Investigation and Mitigation. Special Report 247*, edited by A. K. Turner and R. L. Schuster, 12–35. Washington, DC: National Academy Press.

Spencer, E. 1967. "A Method of Analysis of the Stability of Embankments Assuming Parallel Inter-Slice Forces." *Geotechnique* 17 (1): 11–26.

Sun, J., J. Li, and Q. Liu. 2008. "Search for Critical Slip Surface in Slope Stability Analysis by Spline-Based GA Method." *Journal of Geotechnical and Geoenvironmental Engineering (ASCE)* 134 (2): 252–256.

Taha, M. R., M. Khajehzadeh, and A. El-Shafie. 2010. "Slope Stability Assessment Using Optimization Techniques: An Overview." *Electronic Journal of Geotechnical Engineering* 15 (Q): 1901–1915. http://www.ejge.com/2010/JourTOC15R.htm.

Yamagami, T., and Y. Ueta. 1988. "Search for Non-Circular Slip Surfaces by the Morgenstern–Price Method." In *Proceedings of the 6th International Conference of Numerical Methods in Geomechanics,* Insbruck, Austria, 1219–1223.

Zolfaghari, A. R., A. C. Heath, and P. F. McCombie. 2005. "Simple Genetic Algorithm Search for Critical Non-Circular Failure Surface in Slope Stability Analysis." *Computers and Geotechnics* 32 (3): 139–152.